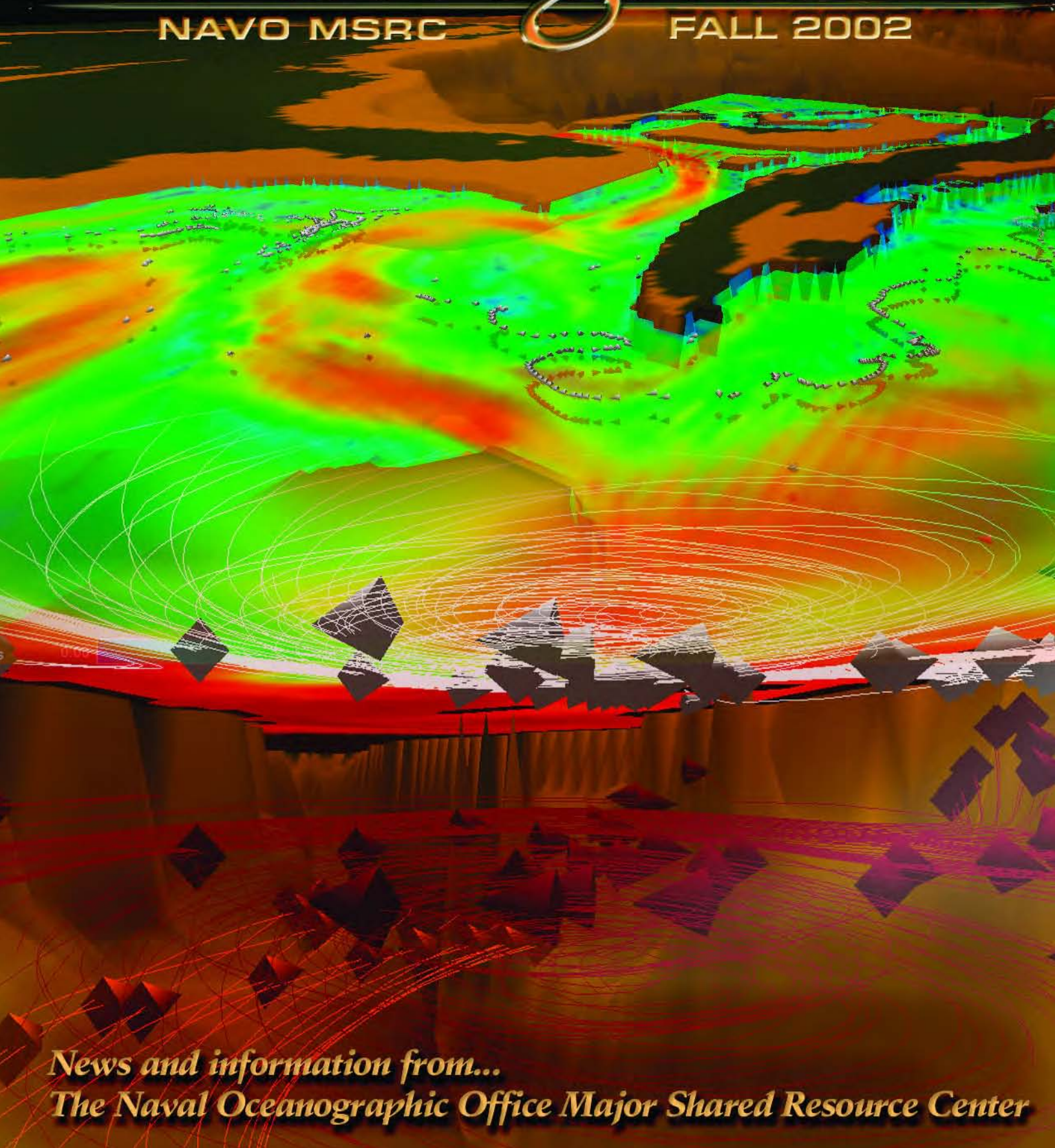
Navigator

NAVO MSRC FALL 2002



News and information from...
The Naval Oceanographic Office Major Shared Resource Center

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2002		**00-00-2002 to 00-00-2002**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
NAVO MSRC Navigator. Fall 2002	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Oceanographic Office (NAVO),Major Shared Resource Center (MSRC),1002 Balch Boulevard,Stennis Space Center,MS,39522**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**28**	

The Director's Corner

Steve Adamec, NAVO MSRC Director

A Decade of Progress

A decade has passed since the DoD HPC Working Group, the predecessor of today's HPC Modernization Office, put to paper and into execution the plans for a DoD-wide HPC environment. At the direction of Congress, that original group of 13 "lucky" folks from the DoD services and agencies faced a monumental task—crafting a vision and plan to leverage, enhance, and unify the service-specific supercomputing programs to meet an overwhelming DoD-wide need for HPC capability. Their efforts produced the DoD HPC Modernization Plan which aggressively cited the need to explore and embrace advanced parallel HPC technology, as well as the need for world-class networking capabilities to tie the nationwide HPC environment and distributed user communities together. If this wasn't hard enough, the group also had to consider the political realities and challenges that are always associated with any program spanning and serving multiple DoD services and agencies.

Only a small handful of the original HPC Working Group remain associated with this program today, and I know we all agree that this program has met and exceeded their original expectations and goals. The progress and benefits to DoD over the last 10 years have been demonstrable and overwhelmingly large and can be measured not only in the cutting-edge nationwide DoD HPC environment but also more importantly by the HPC-enhanced work of thousands of scientists and engineers—work that has made a real difference to the Department of Defense and this nation. Please enjoy this issue of "The Navigator" and accept our sincerest thanks for permitting us to serve you.

About the Cover:

A regional view created from data generated by the Naval Research Laboratory (NRL) Layered Ocean Model (NLOM). Pathlines and colored slice planes show the warm water loop current that provides energy to the hurricanes which migrate to this region. NLOM became an operational Navy model on 27 September 2001.

The Naval Oceanographic Office (NAVO) Major Shared Resource Center (MSRC): Delivering Science to the Warfighter

The NAVO MSRC provides Department of Defense (DoD) scientists and engineers with high performance computing (HPC) resources, including leading edge computational systems, large-scale data storage and archiving, scientific visualization resources and training, and expertise in specific computational technology areas (CTAs). These CTAs include Computational Fluid Dynamics (CFD), Climate/Weather/Ocean Modeling and Simulation (CWO), Environmental Quality Modeling and Simulation (EQM), Computational Electromagnetics and Acoustics (CEA), and Signal/Image Processing (SIP).

NAVO MSRC
Code N7
1002 Balch Boulevard
Stennis Space Center, MS 39522
1-800-993-7677 or
help@navo.hpc.mil

NAVO MSRC Navigator
www.navo.hpc.mil/navigator

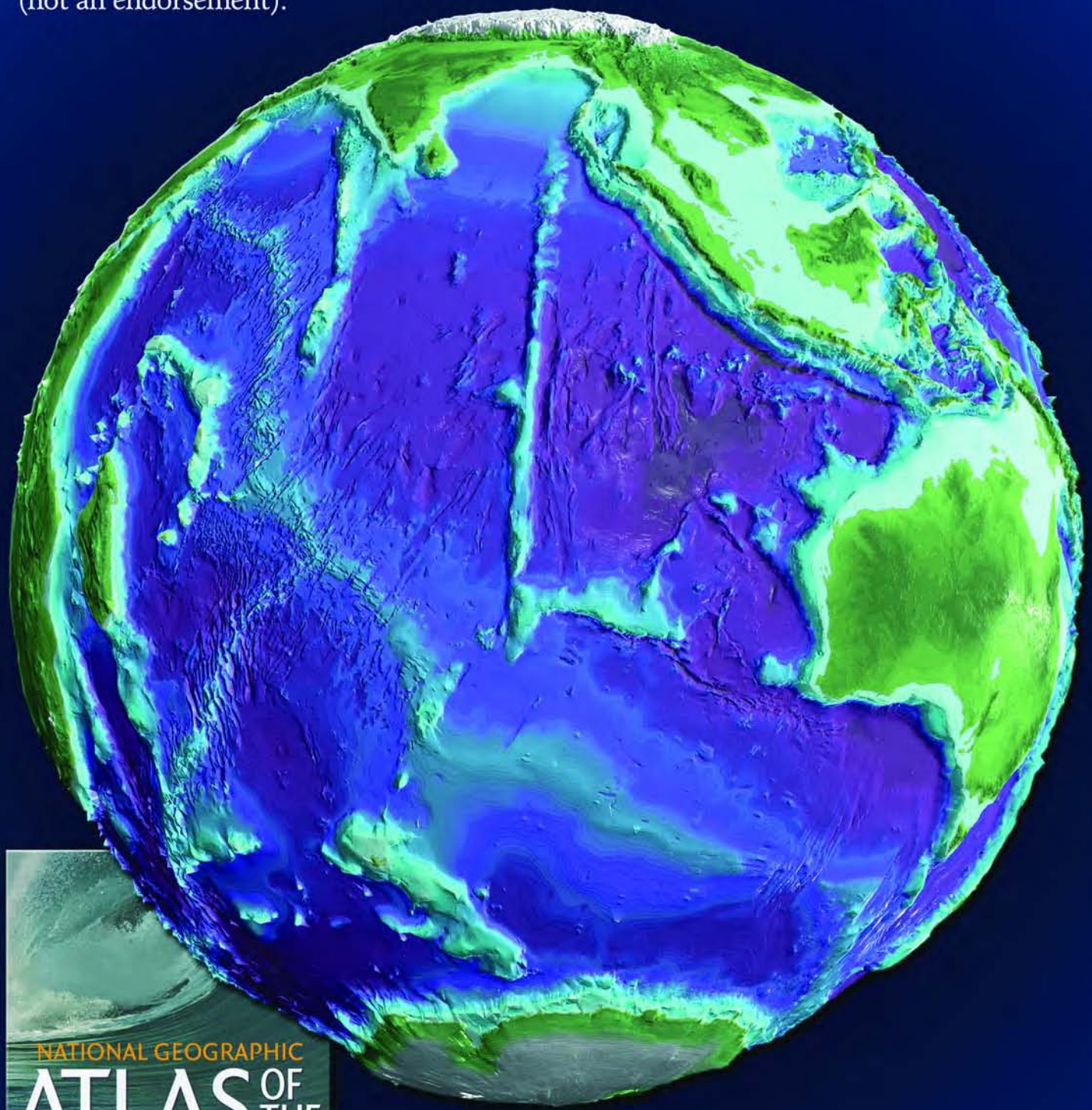
NAVO MSRC Navigator is a biannual technical publication designed to inform users of the news, events, people, accomplishments, and activities of the Center. For a free subscription or to make address changes, contact NAVO MSRC at the above address.

EDITOR:
Gioia Furness Petro, petrogio@navo.hpc.mil

DESIGNERS:
Cynthia Millaudon, cynmill@navo.hpc.mil
Kerry Townson, ktownson@navo.hpc.mil
Lynn Yott, lynn@navo.hpc.mil

Contents

Three-dimensional shaded relief of the Indian Ocean, as it appears on the pages of National Geographic's "Atlas of the Ocean." This image was batch rendered by the NAVO MSRC Visualization Center staff using the MAYA software application (not an endorsement).



THE INDIAN OCEAN

Image generated by the NAVO/DoD MSRC Visualization Center Staff.

COAMPS™ High-Resolution Weather Forecasts for Mississippi and Louisiana Coasts

Dr. Pat Fitzpatrick, Yongzuo Li, Dr. Gueorgui Mostovoi, Mississippi State University Computational Geospatial Technologies Center, Stennis Space Center, MS

Thanks to the recent development of software tools by the National Oceanic and Atmospheric Administration (NOAA) and Unidata, researchers at the Mississippi State University (MSU) Computational Geospatial Technologies Center, Stennis Space Center, MS, have been able to develop and release a powerful nested operational version of the Coupled Ocean/Atmosphere Mesoscale Prediction System™ (COAMPS) that covers the Louisiana and Mississippi coasts by using a Multivariate Optimal Interpolation (MVOI) scheme.

This version of COAMPS runs twice daily at a 14-km and 42-km resolution and displays high-resolution wind forecasts. In developing this version of COAMPS, MSU modelers had to overcome three major hurdles:

- Lack of sophisticated software that can handle continuous transmission of weather data. The Local Data Manager (LDM) software, written by Unidata[1] in Boulder, CO, facilitates the dissemination of weather observation in near-real time and so overcomes this first hurdle.
- Need for weather observations that are in a format easily incorporated into mesoscale models. In the past, this has been difficult, because MSU had to perform much of the post-processing, making data assimilation in real-time weather modeling almost impossible. However, in the past year, the NOAA Forecast System Laboratory (FSL) developed the Meteorological Assimilation Data Ingest System (MADIS)[2], which organizes weather observations into easy-to-use databases in a common format (i.e., netCDF).
- Data quality control so model initial conditions do not become corrupted. Observations sometimes contain bad or incorrectly reported measurements, and these must be removed from the model initialization; however, automating this procedure is difficult. This, normally, is a task requiring the dedication of huge resources that only organizations like FSL can provide. Fortunately, the FSL provides headers to identify probable bad data, thereby allowing MSU programmers to remove these data before they get into COAMPS.

With these new tools, and the support of Navy projects such as the Northern Gulf of Mexico Littoral Initiative (NGLI) and Distributed Marine Environment Forecast System (DMEFS), the High Performance Computing Modernization Program (HPCMP) Programming Environment and Training Program (PET), and the Mississippi Space Commerce Initiative (MSCI), MSU was able to fund a staff of expert mesoscale modelers to write software to handle the COAMPS front end.

Mesoscale initial conditions are provided by using a previous 12-hour first guess of COAMPS that is then updated using optimal interpolation of FSL data. Boundary conditions are provided by either the FNMOC NOGAPS model or the NCEP AVN model. The results are 14-km operational runs, typically available

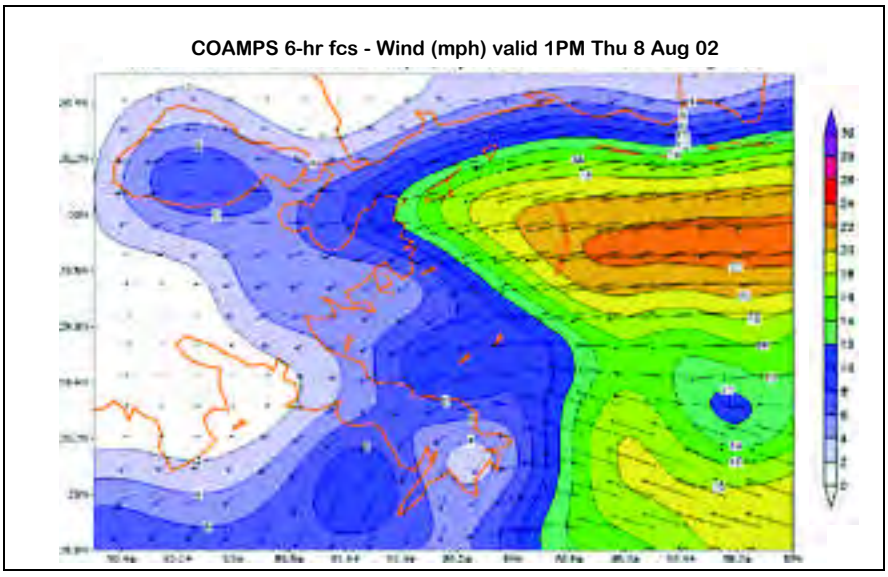


Figure 1. A COAMPS 10-meter 6-hour wind forecast for the Louisiana and Mississippi coasts, initialized 9 August 2002 at 12Z. Note the mesoscale variation captured by COAMPS in the wind field, with weak winds off the Louisiana coast and strong winds off the Mississippi coast, due to a high pressure system moving into the southeastern United States.

by noon and midnight each day, which show high-resolution wind forecasts. These forecasts show a remarkable detail of land and ocean differences, as well as the influence of the land and sea breeze.

An example is shown in Figures 1 through 3 for a 24-hour forecast initialized 8 August 2002 at 12Z (7AM). On this day, the pressure gradient increased due to a high pressure system moving into the southeast United States. As a result, northeast winds increased throughout the day, with winds increasing westward over time.

As shown in Figure 1, winds were initially weak at 18Z (1PM) over Louisiana and the southern coast of Louisiana. In contrast, winds were strong south of Mississippi. With time, these strong winds expanded into Louisiana and its southern coast as shown in Figures 2 and 3.

The model shows this progression, which validates well against buoy and Meteorological Aviation Report (METAR) data. Such information forcasting the big change in wind speed that afternoon would have been useful to boaters off the coasts of Louisiana and Mississippi. Also, note the difference in wind speeds over land compared to water. COAMPS is developed for marine applications, and clearly shows differences between inland and marine winds. These forecasts are being made available to the general public, particularly the maritime community.

The other motivation for the operational runs is the insight they provide in mesoscale modeling research. Typically, parallel runs are performed to study the sensitivity of COAMPS to different physics packages, model resolution, etc., culminating in journal papers such as

Article Continues Page 21...

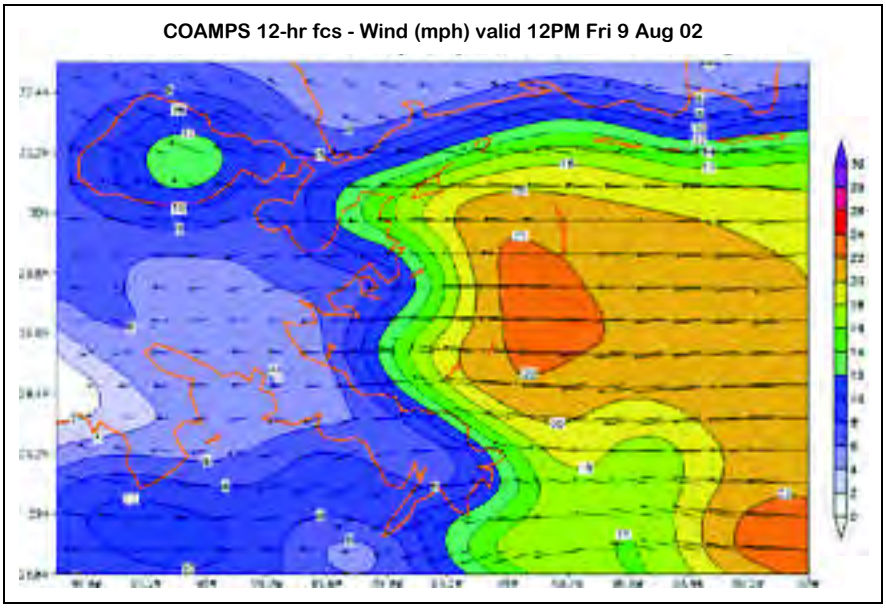


Figure 2. A 12-hour COAMPS forecast, showing the westerly progression of the fast winds into Louisiana.

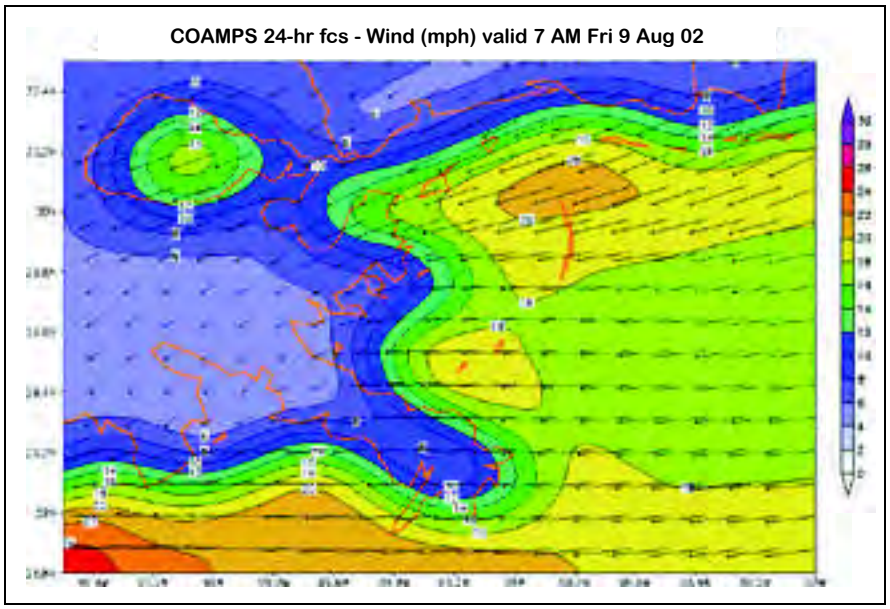


Figure 3. As in Figures 1 and 2, but for a 24-hour forecast, which shows the transition to windy conditions throughout the forecast region. COAMPS' predictions of such wind changes and coastal variations, as well as the distinctly different wind regimes over land and water, should be useful to mariners and coastal residents.

A Distributed Model Coupling Environment for Geophysical Processes

Matthew T. Bettencourt and Shahrdad G. Sajjadi, Center of Higher Learning, Stennis Space Center, MS
Patrick Fitzpatrick, Mississippi State University, Stennis Space Center, MS

In the realm of geophysical modeling the current state-of-the-art models have the capability to run at very high spatial resolutions. This capability has led to a drastic increase in the accuracy of the physics being predicted. Due to this increased numerical accuracy, once neglected effects, such as non-linear feedback between different physical processes, can no longer be ignored.

The ocean's deep water circulation, surface gravity waves, and the atmosphere above can no longer be treated as independent entities and must be considered a single coupled system.

One solution to this problem is to link models together through a series of surface variables. An example would be the evaporative cooling of the ocean, which, at a simple level, requires sea surface temperature, humidity, and temperature of the atmosphere, and would return the mass and heat flux into the atmosphere.

The Model Coupling Environmental Library (MCEL) was developed to simplify the coupling process for models that exchange data at most every time step. Traditionally, model coupling is performed in three ways: file Input/Output (I/O), subroutinization, or Message Passing Interface (MPI).

The traditional way of model coupling is through file I/O as shown by Blain[1] and Hodur.[2] In this case the models are left relatively unaltered and are executed for a very short length of time. Model preprocessors then transform the output files from one model into input files for the second model. Depending on the frequency of coupling, this can be a very costly alternative.

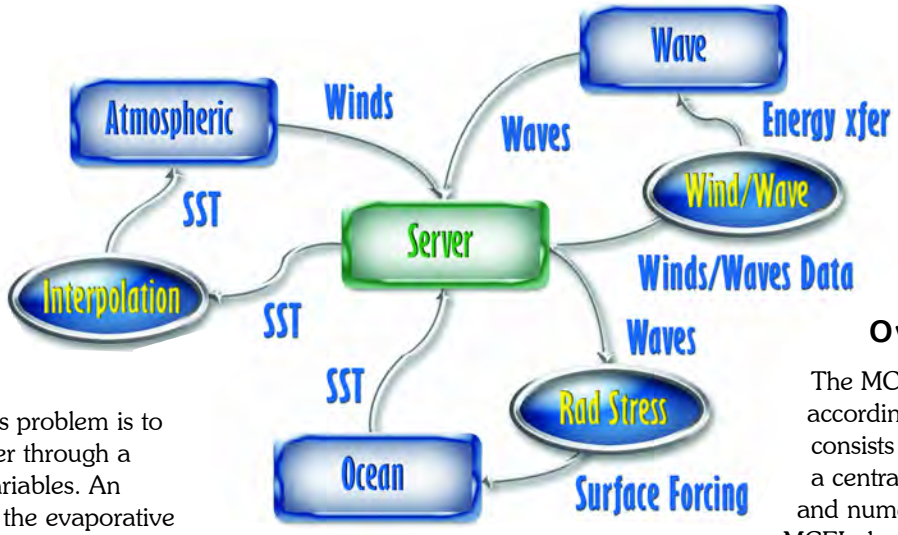


Figure 1. A hypothetical example of a three-model MCEL system.

The second method of model coupling, subroutinization, requires one of the models to be written as a subroutine of the other model. While this can provide the fastest program, it can, however, be quite difficult to implement and maintain such a large multi-physics application.

The final common method for model coupling is through an MPI interface, where calls are added to both applications to send data to each other as demonstrated by Welsh[3] or in an abstract form with the Model Coupling Toolkit.[4]

This approach has the benefit that applications are executed only once, as in the subroutinization method, and the applications are left as independent entities, as in the file-based approach.

However, because MPI uses two-sided communication, it is required that each model be modified explicitly for the set of applications running in a coupled suite.

Overview

The MCEL infrastructure, according to Bettencourt,[5] consists of three core pieces: a centralized server, filters, and numerical models. MCEL, by utilizing a data flow approach, stores coupling information in a single server or multiple centralized servers.

Upon request these data flow through processing routines, called filters, to the numerical models, which represent the clients. These filters represent a level of abstraction for the physical or numerical processes that join different numerical models. The extraction of the processes unique to model coupling into independent filters allows for code reuse for many different models.

The communication between these objects is handled by the Common Object Request Broker Architecture (CORBA). In this paradigm, the flow of information is fully controlled by the clients. Figure 1 represents a hypothetical example of how such a system might be used.

Figure 1 shows three numerical models: Ocean circulation model, atmospheric model, and surface gravity wave model. Each model provides information to the centralized server: sea surface temperature (SST), wave height and direction, and wind velocities at 10 meters above the surface, respectively.

SST is used by the atmospheric model; however, it must first be interpolated onto the atmospheric model's grid. Therefore, the data passes through an interpolation filter prior to delivery. The wave information is transformed into stresses by the RadStress filter using the algorithm by Longuet-Higgins and Stewart.[6]

The final transformation uses the algorithm by Sajjadi and Bettencourt[7] which calculates energy transfer from wind and wave information. The filters represent application-independent processes and can be used to process inputs for any wave or circulation application. With the proper combination of filters and models, arbitrarily complex modeling suites can be developed.

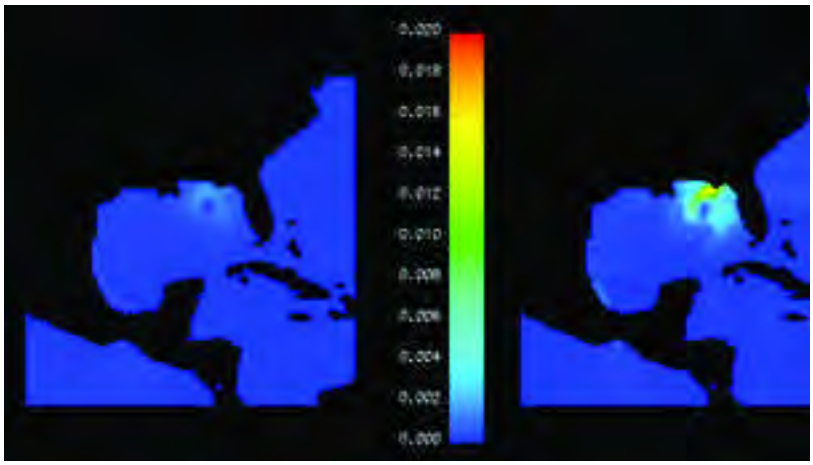


Figure 2. Roughness length for Hurricane Gordon at 9/18/00 22:00. Left: Utilizing Charnock parameterization within COAMPS. Right: Utilizing WaveWatch Parameterization.

Results

The coupling infrastructure has been incorporated into several different models listed below:

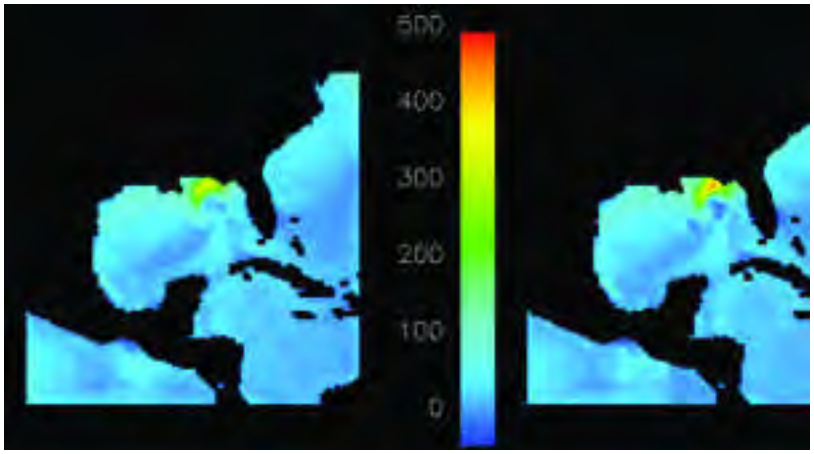
- ADvanced CIRCulation model (ADCIRC)
- Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS)
- Navy Coastal Ocean Model (NCOM)
- REFraction DIFfraction model (REF/DIF)
- WAve Model Cycle 4 (WAM)
- WaveWatch

The work with the COAMPS coupled to the WaveWatch model will be used to illustrate the potential of the coupling infrastructure. COAMPS is a non-hydrostatic atmospheric model that incorporates many physical parameterizations and numerical techniques.

One of these parameterizations is the calculation of the roughness length. COAMPS utilizes the Charnock relationship, which assumes that the waves are in equilibrium with the wind. While this relationship is valid for "old" seas, wind direction and speed changes can throw the system out of equilibrium.

These cases produce much steeper waves and much larger roughness lengths. WaveWatch contains a more sophisticated roughness length approximation that takes into account wave age and produces a more physical roughness length. In the coupling scheme for these two models, COAMPS provided 10-meter wind velocities to WaveWatch every hour of simulation. In return, WaveWatch provided roughness lengths over the ocean.

Figure 3. Sensible latent heat flux for Hurricane Gordon at 9/18/00 22:00. Left: Utilizing COAMPS roughness length calculation. Right: Utilizing WaveWatch Parameterization of roughness length.



Tests of this coupling were conducted on Hurricane Gordon, which struck the coast of Florida on 18 September 2000. The event was chosen because it represented a weak storm, where the WaveWatch roughness length

parameterization was believed to be valid.

The unaltered version of COAMPS under-predicted the intensity of the storm and predicted a path too far to the west of what was actually observed.

COAMPS used 9-kilometer grid spacing over a 121x121 grid with 30 vertical levels. WaveWatch used an 81x81 grid. The model runs were compared between the two-way coupled version versus WaveWatch being forced without feedback.

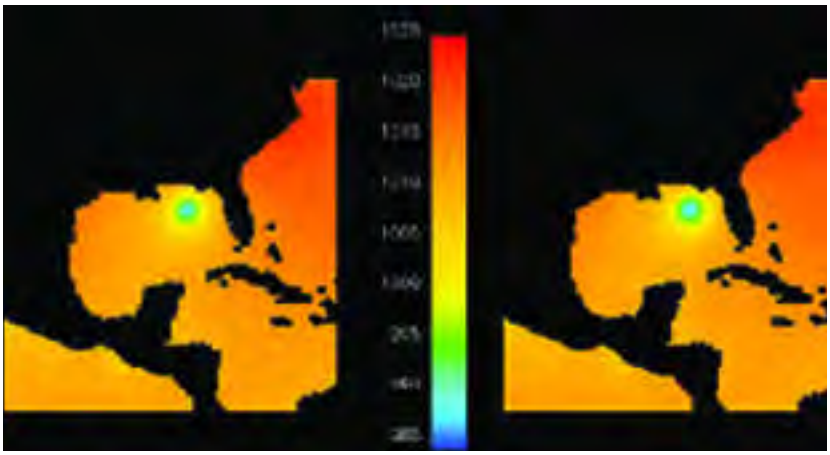


Figure 4. Sea surface pressure for Hurricane Gordon at 9/18/00 22:00. Left: Utilizing COAMPS roughness length calculation. Right: Utilizing WaveWatch Parameterization of roughness length.

Roughness lengths were compared between the two formulations as shown in Figure 2. Over the range of the simulation, the roughness length predicted by COAMPS was typically about ten percent of the value predicted by WaveWatch.

The increased roughness length has two major effects on the storm. First, it increases the kinetic energy transfer from the atmosphere to the ocean. This has a tendency to slow the storm. However, increased roughness length also increases the heat flux to the storm, as shown in Figure 3, which has the tendency to increase the intensity of the event.

These effects combine into a net increase in the storm intensity as seen by the pressure plot in Figure 4. This figure shows a 3-millibar deepening of the pressure at the center of the storm. While this more closely represents what was actually observed with the pressure, it did not improve the track of the storm.

The MCEL infrastructure allows these two models to run concurrently, which can drastically decrease the time until a solution is achieved.

For the problem described above, a one-way coupled mode required 348 seconds per hour of simulation for the COAMPS model and 249 seconds for the WaveWatch model, or a total of 597 seconds. However, in a coupled mode the two jobs could be split onto two different computers, and the solution obtained in 374 seconds, or a speedup of 1.6.

Furthermore, this approach allowed for a more physically accurate solution than the two models running independently. The incorporation of the MCEL resulted in the modification/addition of only a few hundred lines of the two models. This approach simplifies the maintenance of these two models when compared to a single model containing both sets of physics.

Acknowledgements

This work is supported by the Department of Defense High Performance Computing Modernization Office (HPCMO) Common HPC Software Support Initiative (CHSSI) and Programming Environment and Training (PET) programs as well as the Office of Naval Research's supported project Distributed Marine Environment Forecast System.

References

1. Blain, C. and M. Cobb, “Wave-Current Interaction in a Wave-Breaking Environment,” Recent Advances in Marine Science and Technology 2000, 2001.
2. Hodur, R. M., “The Naval Research Laboratory's Coupled Ocean/Atmospheric Mesoscale Prediction System (COAMPS),” Monthly Weather Review, 1996.
3. Welsh, D. J. S., K. W. Bedford, R. Wang, and P. Sadayappan, “A Parallel-Processing Coupled Wave/Current/Sediment Transport Model,” Technical Report PET 00-20, ERDC MSRC Technical Report, The Ohio State University, Columbus, OH, 2000.
4. Larson, E. O. J., and R. Jacob, “MCT - The Model Coupling Toolkit,” http://www-unix.msc.anl.gov/acpi/mct/
5. Bettencourt, M. T., "Distributed Model Coupling Framework," Proc. 11th IEEE Symposium on High Performance Distributed Computing, 284-290, 2002.
6. Longuet-Higgins, M. and R. Stewart, “Radiation Stress in Water Waves,” Deep Sea Research, II:529-562, 1964.
7. Sajjadi, S. G. and M. T. Bettencourt, “An Improved Parameterization for Energy Exchange from Wind to Stokes Waves,” To appear Proc. 2nd Wind-Over-Waves Conference, Ellis Horwood Publishing Co., 2002.

Nanoscale Damage and Dynamic Fracture in Glasses

Cindy L. Rountree, Concurrent Computing Laboratory for Materials Simulations,
Louisiana State University, Baton Rouge, LA
Rajiv K. Kalia, Aiichiro Nakano, and Priya Vashishta, Collaboratory for Multiscale Simulations,
University of Southern California, Los Angeles, CA

Metals and glasses have different fracture mechanisms. Metals are ductile and involve the generation and motion of dislocations during crack propagation. Glasses, on the other hand, are brittle and undergo cleavage fracture without dislocations.

In the past few months, molecular dynamics (MD) simulation studies performed by the Louisiana State University (LSU) group at the Naval Oceanographic Office Major Shared Resource Center (NAVO MSRC) have revealed that, despite these dissimilarities, damage in a glass is akin to that in a metal, albeit at a much smaller length scale. This has just been confirmed by Atomic Force Microscope (AFM) studies.[1]

Molecular dynamics simulations of fracture are performed on amorphous silica (a-SiO2), employing a reliable interatomic potential developed by the LSU group.[2] The potential incorporates steric repulsion, charge transfer, and electronic polarizability of atoms through pair-wise interaction terms. Covalent effects in silica are included through bond-bending and bond-stretching three-body terms.[3] Comparing MD results with various structural, dynamic, and mechanical measurements validates the potential. Over the past decade, the LSU group also developed highly efficient, scalable, and portable multiresolution algorithms to carry out large-scale MD simulations (107-109 atoms) on parallel architectures and algorithms that allow real-time immersive visualization of up to a billion atoms. The fracture simulations described here employ this suite of algorithms.[4,5]

The first set of fracture simulations involves 15 million atoms. Amorphous silica was generated by heating b-cristobalite to 3200K and then quenching the molten system to room temperature.

Subsequently, the system was notched and subjected to a uniaxial strain. Figure 1 (a-c) shows the evolution of damage in the MD simulation. Crack propagation is accompanied by nucleation and growth of

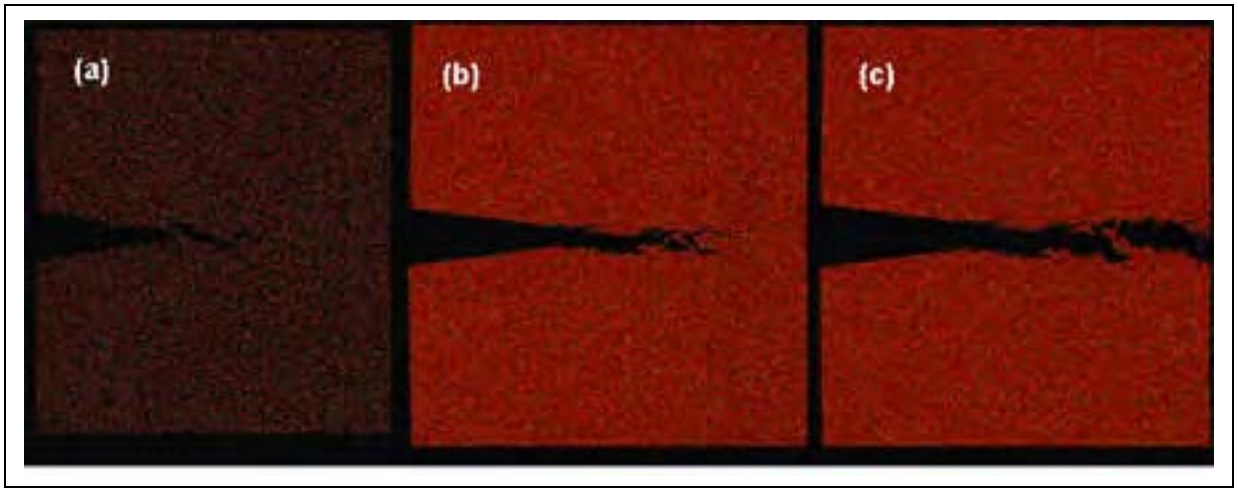


Figure 1. Formation of nanometer scale cavities around the crack tip at an applied strain of 3.2%; (b) a-SiO2 at 6.5% strain; and (c) fractured a-SiO2 system.

nanometer scale cavities up to 20 nm ahead of the crack tip. Cavities coalesce and merge with the advancing crack to cause mechanical failure.

The recent experimental work of Bouchaud and coworkers involving AFM studies of fracture in silica and aluminosilicate glasses also reveals nanocavitation and coalescence of cavities with the crack to be the mechanism of fracture; see Figure 2.[1] Cavitation is a well-known phenomenon in metallic fracture, but the size of cavities is macroscopic.

To make quantitative comparisons with experiments, the LSU group examined the morphology of fracture surfaces by calculating the height-height correlation function,

$$D_{h(r)} = \langle (x(z+r) - x(z))^2 \rangle_z^{1/2} \quad (1)$$

where x is the height of the fracture profile normal to the plane of crack propagation, and $\langle \ldots \rangle_z$ implies an average over z. Fracture experiments by Bouchaud and coworkers reveal that $D_{h(r)} \sim r^{\zeta}$ with roughness exponents $\zeta = 0.5$ and 0.8 below and above a cross-over length, ζ_c ($\sim$ 100 nm), respectively.

The MD simulation finds the first roughness exponent ($\zeta = 0.5$), but the second exponent (0.8) occurs over length scales ($> \zeta_c$) that are inaccessible to a 15-million atom system. Therefore, the LSU group is currently performing a 113-million atom simulation at the NAVO MSRC to map out not only the entire morphology, but also the dynamics of the whole crack front.

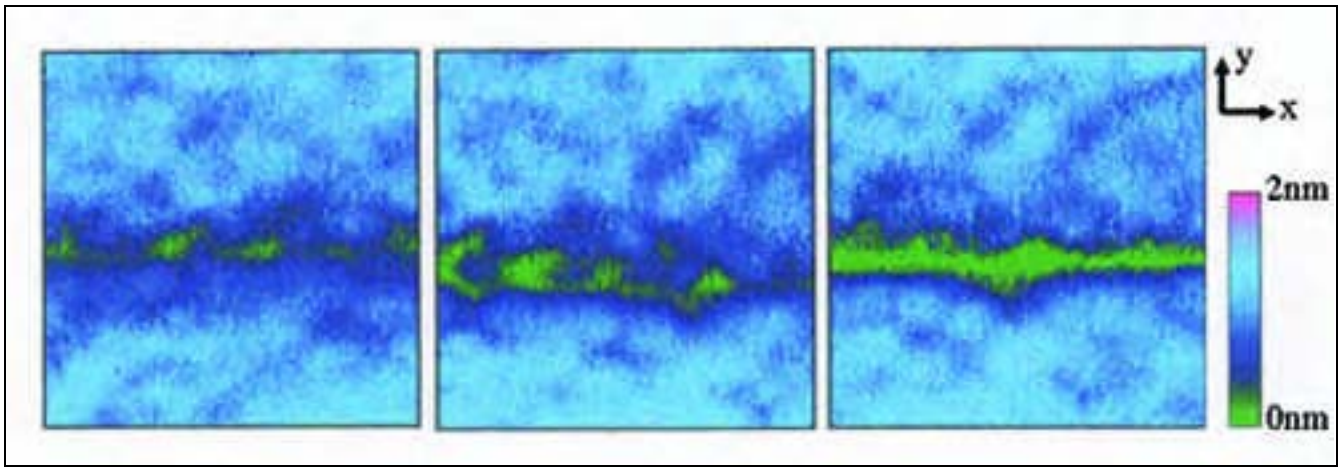


Figure 2. AFM studies of fracture in an aluminosilicate glass also reveal nanocavitation and coalescence of cavities with the crack to be the mechanism of fracture.

References

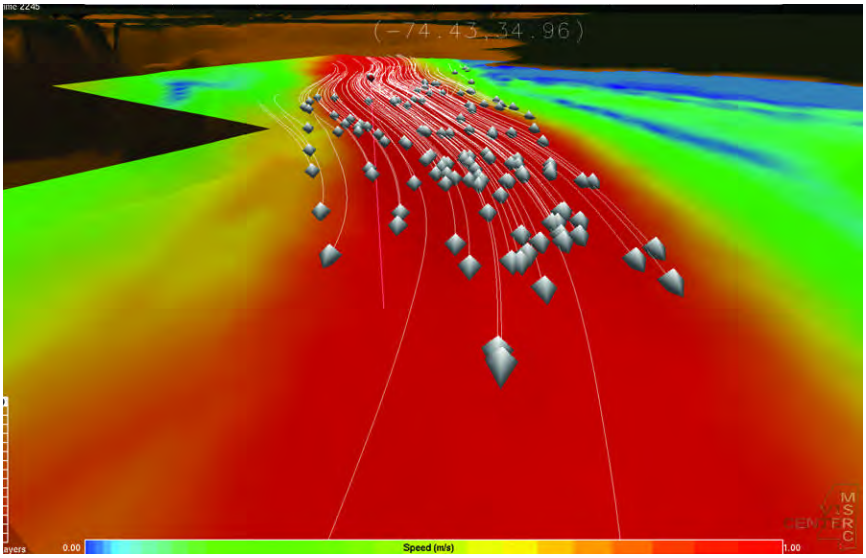
1. Bouchaud, E., private communications.
2. Vashishta, P., R. K. Kalia, J. P. Rino, and I. Ebbsjö, Physical Review B 41 (1990), pp. 12197-12209.
3. Stillinger, F. G., and T. A. Weber, Physical Review B 31 (1985), pp. 5262-5271.
4. Nakano, A., R. K. Kalia, P. Vashishta, T. J. Campbell, S. Ogata, F. Shimojo, and S. Saini, "Scalable Atomistic Simulation Algorithms for Materials Research," Proc. of Supercomputing 2001 (ACM, New York, NY, 2001).
5. Sharma, A., P. Miller, X. Liu, A. Nakano, R. K. Kalia, P. Vashishta, W. Zhao, T. J. Campbell, and A. Haas, "Immersive and Interactive Exploration of Billion-Atom Systems," Proc. of IEEE Virtual Reality 2002 Conference, pp. 217-223.

Data Mining Ocean Model Output: RangerScope

Pete Gruzinskas, Andy Haas, Ludwig Goon, NAVO MSRC Scientific Visualization

One of the Computational Technology Areas supported by the High Performance Computing Modernization Program (HPCMP) is Climate, Weather, and Ocean (CWO) modeling. To this end, state-of-the-art computing architectures are leveraged against the extremely difficult problem of mathematically modeling and predicting the behavior of a variety of ocean climatological parameters.

The problem at hand is that the technology to store, retrieve, manipulate, and display these data has not kept pace with the computational technology. During the last five years, we have seen significant cost reductions associated with applying the status quo in visualization techniques to scientific data sets. This is due in large part to the computer gaming industry, driven by the huge profit margins associated with that market. The scientific community has benefited by these advances in low-cost architectures, but only as a by-product of its original intent, which is entertainment. Even so, these low-cost architectures are not designed to handle the scale of data sizes presented by the scientific community and serve only to make inadequate techniques cheaper to field and use.

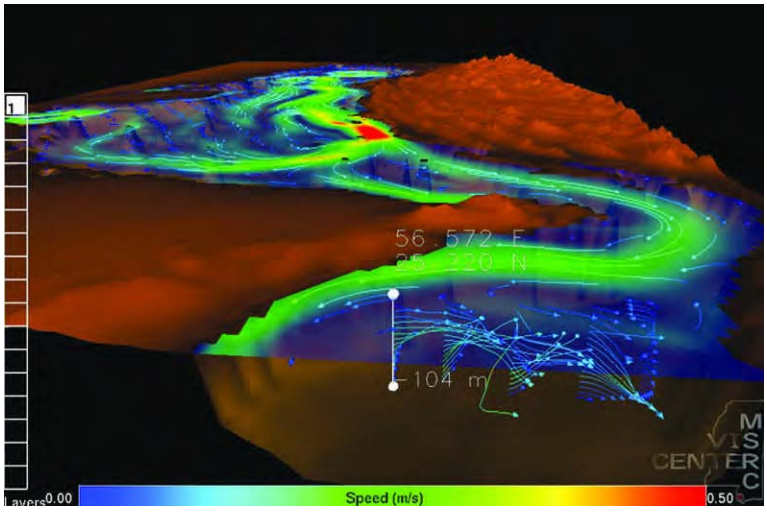


Pathlines of Gulf Stream currents as generated by the Miami Isopycnic Coordinate Ocean Model (MICOM).

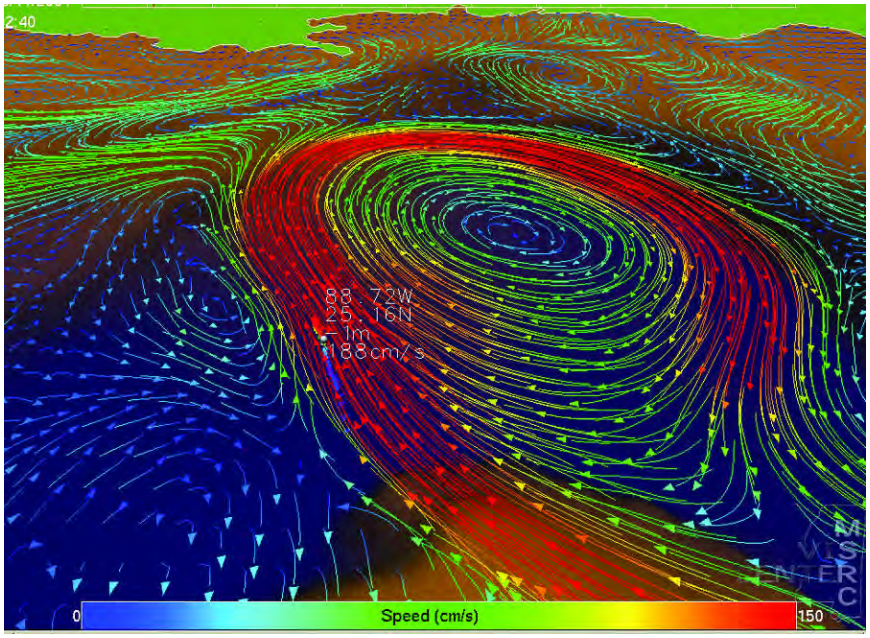
The Naval Oceanographic Office Major Shared Resource Center (NAVO MSRC) Visualization Center is challenged with providing its users state-of-the-art analysis environments for the interrogation of their increasingly large data sets. The data generated by the CWO community involves large domains and high resolutions (either vertically, horizontally, or both) that all vary over time.

This leads to very large data sets (rows x columns x layers x attribute per cell) for each time step and can challenge even the most powerful architectures when trying to extract, or "mine," information from the raw data.

As in most visualization applications, the model output deals with physical parameters that are invisible to the naked eye. This means effective methods of display are required for ocean circulation or currents, sea surface height, temperature, salinity,



Princeton Ocean Model (POM) surface currents in the Persian Gulf.



Gulf of Mexico loop current as generated by the Princeton-Dynalysis Ocean Model (PDOM).

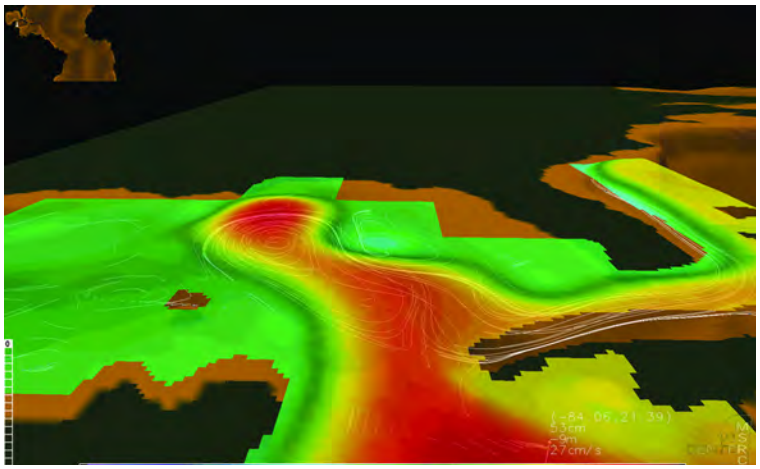
and so on. One analogy, which no doubt started the concept of "data mining," is that the raw data represent a huge block of ore from which gold nuggets of valuable information (features) must be extracted, or mined.

There are two classes of data mining applications: One explores global data in its native format at interactive fashion, and the second applies interactively driven advection analysis to local areas of the model. This article focuses only on RangerScope, an application that satisfies the first mining class. Naval Research Laboratory (NRL) Layered Ocean Model (NLOM), which answers the second class of data mining applications, is examined in the centerfold article of this issue of The Navigator.

RangerScope is a general-purpose tool for mapping sequences of large two-dimensional data files onto an optional three-dimensional terrain elevation. The tool enables a user to roam across the field of data while it is played back at interactive speeds. The data are kept in a sequence of netCDF files, and any 2-D variable data set present inside the files is eligible for viewing by clicking the mouse.

RangerScope is able to render very large data arrays because it uses dynamic level-of-detail algorithms. These algorithms enable the user to travel real-time anywhere in the field and see information at its native resolution. The algorithms have common applications across all cases where large data arrays need to be managed for interactive exploration.

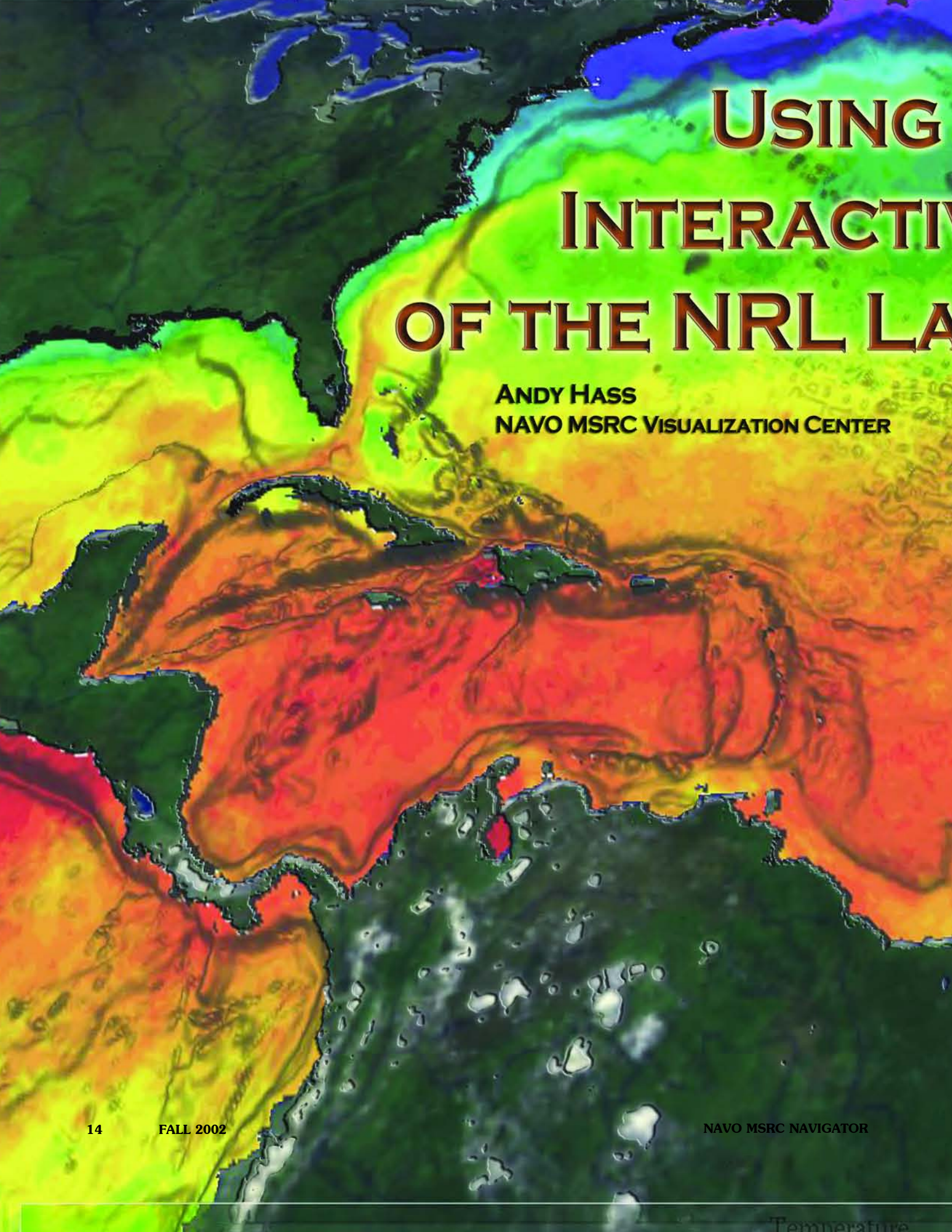
Remote rendering will play a critical role in the dissemination and analysis of high-resolution model output. The NAVO MSRC Visualization Center staff will continue to evaluate technologies that reduce the limitations on analysis created by physical distance. This includes collaborative or data-sharing technologies that allow disparate groups or individuals to view and analyze the same domain simultaneously.



Gulf of Mexico sea surface height generated by MICOM.

Acknowledgements

This article is an abstract of the paper "Data Mining Ocean Model Output at the Naval Oceanographic Office Major Shared Resource Center" presented at the MTS/IEEE Oceans 2002 Conference. The full text of the article can be found at: http://www.navo.hpc.mil/Vizlab/papers.html.



Using Interacti of the NRL La

Andy Hass
NAVO MSRC Visualization Center

NetCDF for ve Visualization ayered Ocean Model

The Naval Research Laboratory (NRL) Layered Ocean Model (NLOM) is a real-time eddy-resolving global ocean nowcast/forecast system that has been running at the Naval Oceanographic Office (NAVOCEANO) since 18 October 2000. NLOM became an operational model on 27 September 2001.

The daily runs of the nowcast/forecast system are performed on the RS/6000 SP3 (HABU) IBM SP, with 30-day forecasts run once a week on the IBM SP.

NLOM operates at 1/16-degree resolution with lateral boundaries, which follow the 200-meter isobath. It has six isopycnal layers, plus mixed layer and realistic bottom topography, which are confined to the lowest layer of the model.

The output variables are layer depth, layer velocity, surface temperature, and surface height. The size of NLOM (4096x2304x7)makes it a challenge to interactively seek and visualize data spread across large areas of the grid.

Output Format

Historically NLOM output was stored with each component and layer of data archived in separate files, with the file data kept in unformatted binary floating-point format. Direct visualization of this unformatted information spread across many files was difficult to achieve, especially when an interactive, real-time three-dimensional (3D) rendering of the information was desired. Current methods of viewing the model's output data have been through image plots and animations made from National Center for Atmospheric Research - Graphical Kernel System (NCAR-GKS) based software.

Article Continues...

To improve the portability and accessibility of model data across different hardware and software, netCDF files are now used to store the variables of NLOM data.

Each netCDF file contains a list of all variables in a given time snapshot (currently 24-hours apart). In addition to the data variables, a netCDF file contains a regularly spaced longitude/latitude (X,Y) grid, which is listed as one-dimensional X and Y variables in the file. The depth (Z) grid uses 6 isopycnal layers to represent the stratification of water density. The isopycnal nature varies the thickness of each Z layer across the grid.

Because the thickness changes over time, each layer has a Z data variable that contains the depths at a given (X,Y) at that layer. Variables of U and V velocities are kept for each layer as well.

Lastly, two variables of surface temperature and sea-surface height are also output for a total of 20 two-dimensional variables (6 U's, 6 V's, 6 Z's plus 1 temperature and 1 height). All 20 variables are labeled and written into a netCDF file named "UVZTH.yyyy.mm.dd.CDF," where "yyyy," "mm," and "dd" indicate the date of model run.

The NAVO MSRC Visualization Center staff worked with the NRL group to set in place a system that transfers the UVZTH files to the NRL SGI Onyx 3400 Fiber Channel RAID storage, which acts as a central repository from which all model runs up to the previous day can be accessed.

Visualization and Interaction

The NAVO MSRC Visualization Center developed the NLOMExplorer and RangerScope interactive visualization tools that operate directly on the NLOM netCDF files. These tools access the UVZTH files via the standardized netCDF interface and have been engineered to operate on the data immediately after they have been read so that on-the-fly interaction is possible.

NLOMExplorer was the first application developed for NRL. The application allows the user to fly across and into or out of the model bathymetry. While doing this, the user can place particles in the field of view with a mouse click. The particles may be added at any number of selected depth layers in the model.

The application interpolates the daily sequence of UVZTH files into a continuous four-dimensional space-time domain. The particles are advected as time is interpolated through the file sequence. The number of files in the sequence does not affect the performance (i.e., the application performs the same while running with 4 files or 400 files).

NLOMExplorer divides the model's area into an array of data blocks. As each particle is added, the application determines in which block of data it resides. Data Input/Output (I/O) is performed only on those blocks that contain particles.

As particles flow from block to block, NLOMExplorer frees memory, graphics, and I/O resources from those blocks not in use. This allows the user to explore a domain much larger than the resources of a desktop computing system can accommodate as a whole. The data-blocking algorithm is integrated with the standardized netCDF access library.

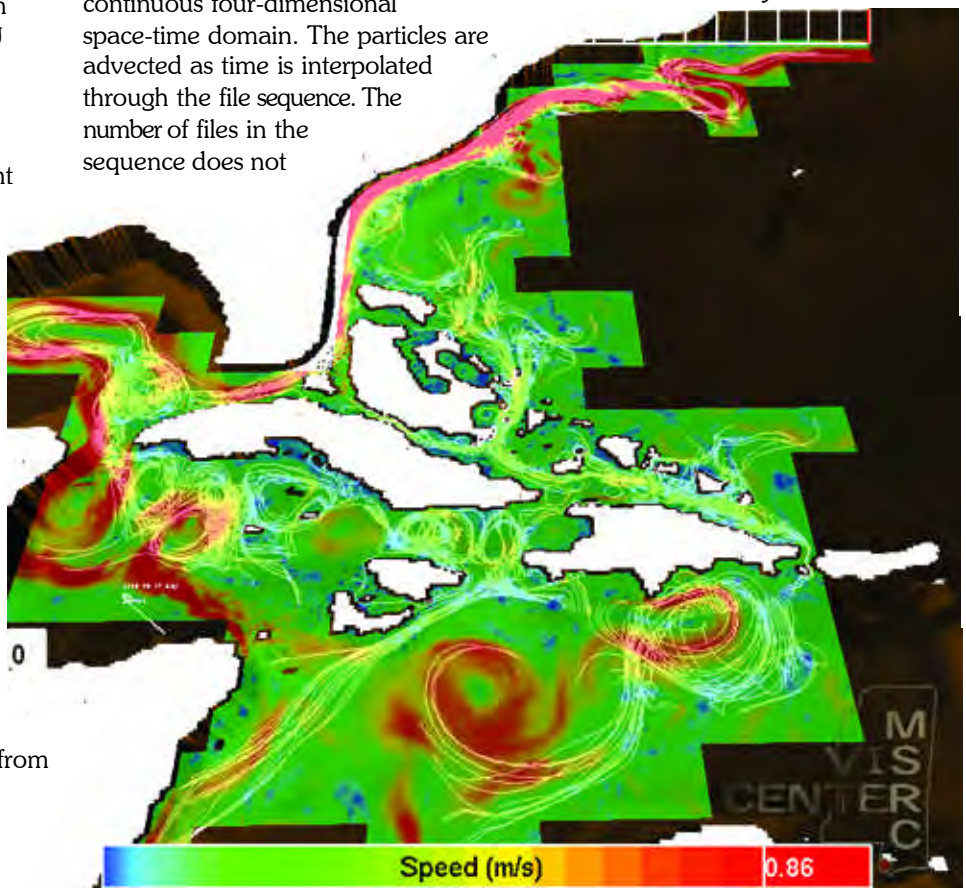


Figure 1. Snapshot of NLOMExplorer output. The user is viewing particles advecting around the Caribbean. The white path lines trace the particles' flow. The data blocks that contain particles are colored by the speed of the block's velocity data.

In contrast to NLOMExplorer, RangerScope is an application for exploring large two-dimensional (2D) variables directly from netCDF files. RangerScope is a general-purpose tool that gives the user the ability to roam across any netCDF variable containing real data. Data can be selected on the fly, where it is colored, mapped, and animated over an optional 3D-rendered terrain. The terrain is a separate variable of Z data that is triangulated and rendered on the fly.

The level-of-detail rendering engine can render an 8000x4000 grid at interactive frame rates on 1-GHz PCs with nothing more than a $200 graphics card. The engine works in parallel with a shared-memory process that reads a data variable's contents into memory. The engine maps and colors the data to the grid.

Grids in RangerScope are 2D and can only be rectilinear. Grid information inside a netCDF file is given by the existence of two one-dimensional (1D) variables.

If the dimensions of both variables equal the number of columns and rows of the data, then the 1D variables' values are used as the X and Y grid. If the grid is meant to be regular, it is given as a list of uniformly spaced X and Y values. If no 1D grid variables are found, RangerScope uses the data variables' I and J indices as the X and Y grid values.

The terrain and data variables are given by different netCDF files, so each can have grids with different locations and resolutions. If a terrain is given, RangerScope correlates the data's grid position to the terrain.

RangerScope runs only on Unix/Linux because it uses multi-processing parent/child processes that communicate via shared memory. The parent/child relationship allows the child to read a data variable into shared memory without interrupting the main parent process. As the parent is rendering the new data, the child is simultaneously reading the variable's data from the next file in the sequence.

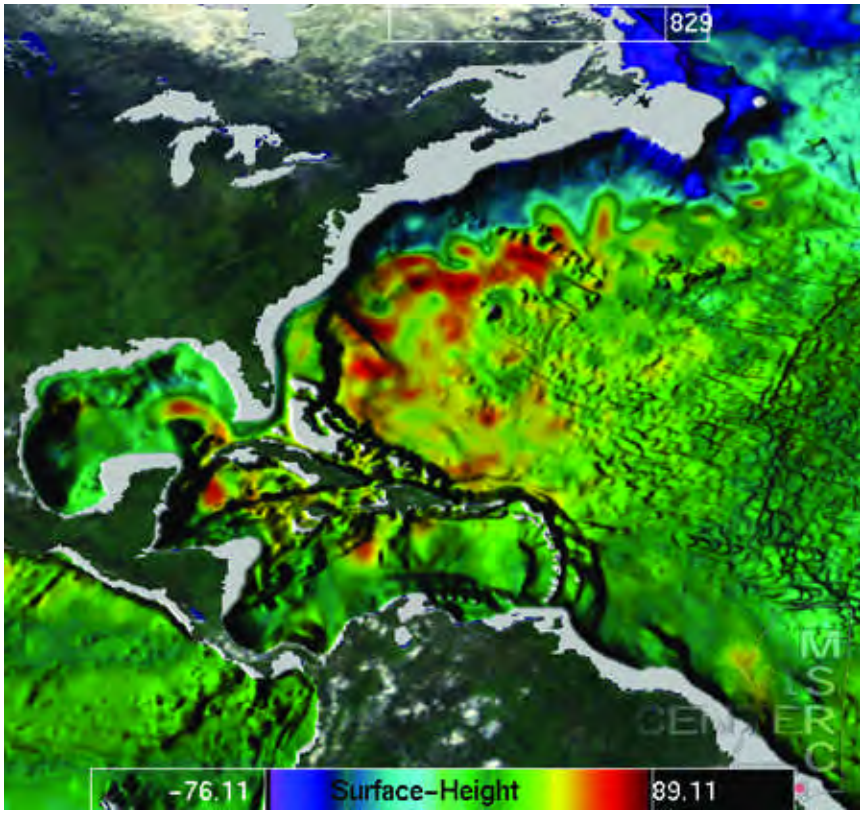


Figure 2. A live snapshot of RangerScope output. The sea-surface height layer is draped over the grid's bathymetry. This can be used to correlate behavior at the sea surface with the terrain beneath.

Conclusion

By using netCDF, NRL is able to maintain up-to-date daily information on the NLOM model runs. The visualization applications presented utilize the netCDF library to access this information in a portable, standardized fashion.

At this time, NRL is using NLOMExplorer and RangerScope with its current NLOM model files. The system to create these files is already in place and automatically produces them on a daily basis. Therefore, when either of these two applications or a third-party application is invoked, NRL researchers can access both high-end interactive visualization and conventional viewing on their desktops.

The goal in selecting the netCDF format was to couple interactive analysis with the common data format. The netCDF format represents an optimal standard format for visualization applications, both custom and third party, but more importantly can accommodate critical attributes that make data retrieval and subsequent display far more efficient.

MARCELLUS Has Arrived

David K. Magee, High Performance Computing Systems Engineer

The computational power of the Naval Oceanographic Office Major Shared Resource Center (NAVO MSRC) has been upgraded by the installation of its newest Terascale HPC System, an IBM RS/6000 POWER 4 system. The system, named MARCELLUS, provides a peak computational power in excess of 6.2 trillion calculations per second, making it the sixth most powerful computer in the world.

MARCELLUS is based on 1,184 IBM POWER4 1.3-GHz processors, with 1,408 terabytes (TB) of distributed memory and 24 TB of direct attached IBM disk arrays. The addition of the POWER4 system brings the aggregate computational capability of the NAVO MSRC to an excess of 8.3 trillion operations per second.

For the first time in a system of this size, the NAVO MSRC allowed the POWER4 to be assembled on-site. Figures 1 through 4 provide a snapshot of the assembly processes in the Operations Center. All the processes surrounding assembly, testing, and integration went well, and early access users report significant increases in the performance of their applications.

MARCELLUS Step 1: Some assembly required.

MARCELLUS Step 2: Layout and configuration.

MARCELLUS Step 4: Installed and ready.

MARCELLUS Step 3: Assembly and installation.

While the architecture of the POWER4 and the POWER3 systems are similar, there are distinct differences. The POWER4 system is divided into 148 eight-way nodes. Of these nodes, 144 have 4 gigabytes (GB) of memory, while the remaining four nodes have 64 GB of memory. These four nodes facilitate users who have relatively small processing needs, yet have large memory requirements.

The net result of the installation of MARCELLUS is a significant addition to existing resources that provides DoD scientists and researchers the capability to run the largest DoD Challenge applications. The applications run on the MARCELLUS POWER4 system will allow the construction of greater detail in models of ocean waves, currents, and temperature than ever before. This detail will allow greater insight into ocean behavior, which will enable scientists to better predict ocean behavior with a precision unimaginable five years ago. This precision will improve search and rescue capabilities globally and increase the safety of naval vessels and commercial shipping.

Porting Applications from HABU to MARCELLUS

Dr. John Cazes, Challenge Project Support, NAVO MSRC

With the arrival of MARCELLUS, the new NAVO MSRC 1184 processor IBM eServer Cluster 1600, NAVO MSRC users will have access to one of the largest IBM pSeries 690 systems in existence. For users transitioning to MARCELLUS from HABU, the NAVO MSRC 1336 processor IBM SP3, we will discuss a few minor changes to their batch scripts and compiler arguments so they may take advantage of the enhancements offered by MARCELLUS. (See MARCELLUS Has Arrived, facing page) for an introduction to MARCELLUS.)

BATCH SCRIPTS

The design of MARCELLUS is very similar to that of HABU. As with HABU, MARCELLUS is a cluster of Symmetric Multi-Processor (SMP) nodes connected through a dedicated high-speed switch running the IBM AIX operating system. As on HABU, LoadLeveler is the batch queuing system. As a result, most batch scripts from HABU should work with little trouble on MARCELLUS as long as allocations are present and queue limits are observed. There are just a few batch script changes that should be made to account for configuration differences between the two machines.

MARCELLUS is configured with eight-way SMP nodes rather than four-way SMP nodes as on HABU. Therefore, to specify the correct number of required processors in your batch script, the LoadLeveler directives, node, and `tasks_per_node` should be modified to reflect the availability of eight instead of four processors per node. This would also apply to the number of threads initiated by OpenMP codes or any other multi-threaded applications.

For example, if your Message Passing Interface (MPI) application requires 32 processors, on HABU you would have requested 8 nodes with 4 tasks per node. However, on MARCELLUS, you should request 4 nodes with 8 tasks per node. Otherwise, your application will still run on 32 processors spread across 8 nodes, but your allocation will be charged for 64 processors instead of 32.

In addition to having larger SMP nodes than HABU, MARCELLUS is configured with a dual-plane switch rather than a single-plane switch. The dual-plane SP2 switch utilizes two completely separate switch networks with two switch interfaces per node. This allows communications to be striped across both networks for greater bandwidth than a single-plane switch, such as the Colony switch on HABU. While on HABU the `network.mpi LoadLeveler` directive is set to `css0` or `switch`, it should be set to `csss` on MARCELLUS to enable striping across both switch interfaces. If the `css0` or switch setting is used on MARCELLUS, the application will communicate with only one interface, using approximately two-thirds of the available bandwidth. Also, it would be very unwise to leave this LoadLeveler directive out for an MPI program, as the default network setting is ethernet.

COMPILER ARGUMENTS

MARCELLUS utilizes the new 1.3-GHz POWER4 microprocessor in place of the 375-MHz POWER3 processor on HABU. This latest generation of POWER4 processors from IBM incorporates two processors and a shared Level 2 cache on one chip. Although the cache structure and chip interface have changed from the POWER3 architecture, the POWER4 processor is backwards compatible with all 32-bit binaries compiled for the Power3 processor on HABU. However, 64-bit binaries compiled on HABU under the AIX 4.3 OS are incompatible with the AIX 5.1 OS on MARCELLUS. As a result, all 64-bit binaries and object code must be recompiled for MARCELLUS. This will change later in the year when HABU is upgraded to AIX 5.

Even though most applications will run on MARCELLUS without recompiling, for speed all Fortran code should be recompiled to tune the application for the p690 architecture. This may be done by recompiling with the `-O4` flag or with the `-O3`, `-qarch=pwr4`, and `-qtune=pwr4` compile flags.

Unfortunately, the IBM C/C++ compilers have not yet been updated to take advantage of the Power4 processor. Consequently, for C/C++ applications, the architecture specific flags should remain at pwr3 until further notice.

With the newer parallel environment on MARCELLUS, there is now support for 64-bit MPI libraries. So, if you are one of the many users who have had to compile and link with the `-bmaxdata:<bytesize>` flag to increase your data space above the default 256-MB segment, life just got easier. You can now compile and link with the `-q64 flag` to address more than 256 MB or even over the 2-GB limit. But, you must compile with the threaded libraries. As with the MPI2 features, the 64-bit MPI libraries are available only when compiled with the thread-safe compilers. Also, as on HABU, MARCELLUS has 1 GB per processor or 8 GB per node for all but four high-memory nodes.

Article Continues Page 26...

NAVO MSRC PET Update

Eleanor Schroeder, NAVO MSRC Programming Environment and Training Program (PET) Government Lead

It seems the most frequently asked question these past few months is what is PET doing? It's a good question—and for this Navigator issue, I'll do my best to let you know what Component 1 of PET is doing.

As a reminder, Component 1—located at the Naval Oceanographic Office—is primarily responsible for the computational technical areas Climate/Weather/Oceanography (CWO) and Environmental Quality Modeling (EQM), as well as for the cross-cutting area of Computational Environments (CE).

Our CWO on-sites, Dr. Tim Campbell at NAVO MSRC and Dr. Phu Luong at the U.S. Army Engineer Research and Development Center (ERDC), have been very busy over the past year. These on-site representatives provided technical assistance to our HPC users, especially those working with SWAN, ADCIRC, QUODDY, WISWAVE, and CH3D-Z. An Introduction to Scientific Visualization class was held in the summer, with emphasis on scientific visualization tools and applications pertinent to the CWO community. We are also happy to have on board Dr. John Romo, who will be our much needed on-site at Monterey.

The EQM on-site, Dr. Jeff Hensley at ERDC, has also been quite instrumental in assisting the EQM community. He has worked on OpenMP improvements to CE-QUAL as well as provided technical assistance and improvements to several of the RMA codes, UTPROJ, and FEMWATER. Some initial conversations have begun with the EQM folks at the Space and Naval Warfare Systems Command (SPAWAR) in San Diego as well. We've also been fortunate to have a good team of people at the University of Texas in Austin who have been diligently working on EQM models, such as CE-QUAL-ICM and ADH, as well as investigating methodologies using the discontinuous Galerkin method.

In Continuing Education (CE), one of the major achievements from the first year of the new PET program was the successful deployment of PAPI 2.1 at the four Major Shared Resource Centers (MSRCs) as well as at Maui High Performance Computing Center (MHPCC) and Arctic Region Supercomputing Center (ARSC). This was the first time a software tool was deployed at multiple Shared Resource Centers (SRCs) in a consistent manner. It is the start of what will hopefully be a process by which other software tools can be consistently deployed to the SRCs. The process is being fully documented and will eventually reside on the PET Online Knowledge Center. The NAVOCEANO on-site, Dr. Tom Cortese, has been quite busy with beta testing several tools that may be of potential use to the Department of Defense High Performance Computing community.

The PET classrooms received another nice gift this past summer (many thanks again to the MSRC). Two new BARCO projectors were installed—these projectors are much smaller than the old ones and are quite an improvement! The classrooms have been used quite a bit for classes sponsored by PET, the NAVO MSRC, NAVOCEANO, and National Oceanic and Atmospheric Administration the past few months. We appreciate the assistance provided by both government and contractor staff in helping the PET classroom facilities remain state of the art.

PET also held its first summer intern program from June to August of 2002. My description of our successful first year follows.

PET Summer Interns

The PET summer intern program was a great success at the NAVO MSRC. This year two students participated and completed the program: Joel Konkle-Parker, working on his degree in Aerospace Engineering from Mississippi State, and Nicholas Green, who is working on his Associate in Science from Florida Community College.

Joel Konkle-Parker worked with his mentor Terry Jones, Senior Systems Integrator of Northrop Grumman Information Technology (NGIT). Under his guidance, Joel participated in several aspects of the integration and acceptance of the new IBM RS/6000 Power4 system.

Joel's first real goal was to assist NGIT with the third-party software requirements for the new IBM. He also participated in the update of the Expansion and Analysis (E&A) web site and various statistical analysis tasks, such as keeping the Resilient Mass Storage Server (RMSS) utilization records up to date, creating graphs of the System Activity Rate (SAR) data for the various systems, putting together presentations detailing the results of the analysis, and performing other tasks.

COAMPS...continued from page 6

the soon-to-be submitted article "Validation of Coastal Wind Forecasts - A Sensitivity Study of COAMPS 2.0 at 9- and 27-Km Resolution" to the Monthly Weather Review.

Other research includes the development of a coupling scheme between wave models and COAMPS. This PET project has funded applied research toward incorporating the NCEP wave model, WaveWatch, into COAMPS, and improving wave growth parameterization physics.

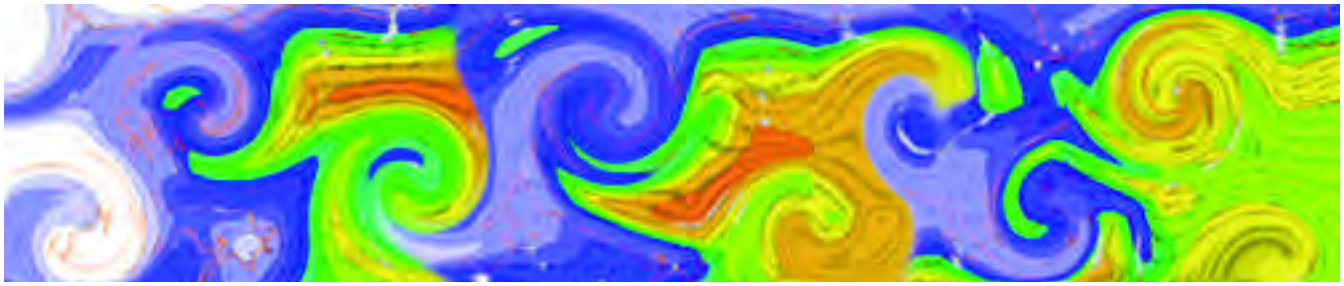
References

1. www.unidata.ucar.edu
2. www-sdd.fsl.noaa.gov

Contacts

To access the forecasts generated by COAMPS, go to http://www.ssc.erc.msstate.edu/NGLI/coamps_DA/coamps_new.html.

Dr. Pat Fitzpatrick, fitz@erc.msstate.edu, or 228-688-1157.



Joel said of his experience, "Not only was I busy most of the time, but I had an interest in what I was doing and felt that I was contributing something to the group. Thank you for the wonderful summer, and I will immediately recommend your center to anyone looking for an internship in that field."

Nicholas Green worked with his mentor Randy Becnel, Lead of the NAVO MSRC Network Group of NGIT. His internship work dealt with the troubleshooting of dial-in problems, making and installing a cat-5 or fiber gigabit Ethernet cable, and configuring V-LAN and V-LAN Trunking ports on the switches.

Nicholas also had a chance to configure and install three Cisco 2900 XL switches (along with troubleshooting on the Crays and SGIs), to participate in the crash test of the IBM SP4, and to help install the Voice-Over Internet Protocol (IP) telephone system at the NAVO MSRC.

Nicholas said of his experience, "I feel that the summer internship was very informative and definitely beneficial to me. It gave me the opportunity to see what working in an HPC center is like. I was exposed to and given the opportunity to work on many types of networking-related materials. Throughout the entire internship I felt as though I accomplished a lot."

Feedback from Terry Jones, mentor to Joel, was very positive. "The intern program helped us achieve goals that we would have otherwise been unable to achieve. Having such a high-caliber individual as Joel participating made the experience rewarding and one that far exceeded my expectations. If the opportunity presented itself again, I would eagerly participate again, and would definitely like to have Mr. Konkle-Parker work with us."

Equally positive was the feedback received from Randy Becnel, Nicholas' mentor. "Nicholas was a valuable contributor to the NAVO MSRC mission during his internship. As lead of the networking group, I found the intern program to be extremely beneficial to both the intern and NGIT's mission at the NAVO MSRC and would welcome the opportunity to participate in future intern programs."

We look forward to the 2003 Summer Intern Program. We will be looking for mentors again—the sooner you let us know you're interested, the better able we are to match an intern to your needs.

THE PORTHOLE

Stan Harvey, NAVOCEANO; CAPT Hugo Gorziglia (Ret), IHO Board of Directors/International Advisor to Chilean Hydrographic and Oceanographic Service (SHOA); Paul Cooper, NAVOCEANO; Dave Cole, NAVO MSRC; CAPT Fernando Mingram, Director, SHOA; Dr. Mario Caceres, Oceanographic Department, Division Sub-Director, SHOA; LCDR Patricio Carrasco, Research and Development Department, Division Sub-Director, SHOA; and Eric Villalobos, NAVOCEANO.

Steve Adamec, Director, NAVO MSRC (center), accompanied by spouses of NAVOCEANO personnel.

Jack Tamul, NAVOCEANO; LT Pak and LT Moh, Republic of South Korea Navy (ROKN); and Dave Cole, NAVO MSRC.

Government Services Administration (GSA) visitors: Ron Holland, Sandy Bates, Bob Suda, Roz Fullerton, and John Mayes.

Ludwig Goon, Scientific Visualization, demonstrates Multichannel Sea Surface Tempurature (MCSST) to teachers involved in the NASA Summer Program.

RADM Larry Baucom (Ret), Director, Geospatial Systems Integration and Homeland Security (HLS) Naval Facilities Engineering Command; Dick Bilden, Naval Facilities Command, Stennis Space Center; and Dave Cole, NAVO MSRC.

Principal Investigators for Effects and Sound on the Marine Environment (ESME) and Synthetic Natural Environments (SNE).

Right:
CAPT Philip G. Renaud, Commanding Officer NAVOCEANO; Dr. Edward Johnson, Technical Director, NAVOCEANO; RADM Thomas Wilson, Oceanographer of the Navy; and Steve Adamec, Director, NAVO MSRC.

Right:
Tom Crew, NAVOCEANO; Sally Garrett, Defence Technology Agency, New Zealand Defence Force; Dave Cole, NAVO MSRC; and Dr. John Kay, Defence Technology Agency, New Zealand Defence Force.

Patricia Walker, Deputy Assistant Secretary of Defense for Materiel and Facilities, Office of Reserve Affairs; CAPT Philip G. Renaud, Commanding Officer, NAVOCEANO; and Dave Cole, NAVO MSRC.

Eric Wamble, Senator Trent Lott Senior Staff; Dr. Ed Johnson, NAVOCEANO Technical Director; Paul Meyers, NAVO MSRC; and Pete Gruzinskas, Scientific Visualization.

Decreasing Batch Queue Wait Time on the IBM SP3 (HABU)

Jared Barousse, NAVO MSRC User Support

Waiting too long for your job to run in batch? You just might be. Our intention in this article is to demonstrate how to achieve a quicker turnaround for batch jobs submitted to the IBM supercomputer HABU using available system commands and manipulation of the user's job script.

We will focus on two keywords in a job command file for the purpose of this article. The "#@ wall_clock_limit =" and "#@ node=" are two important factors in the scheduler's backfill algorithm. The requests made by these keywords tell the scheduler where your job can fit in the runtime queue. If the "#@ wall_clock_limit =" request is not made, then the default maximum time for the class is made.

Depending on the number of nodes requested, a job could take hours before ever going into a run state, a situation that could have be avoided if a shorter limit was simply requested.

Generally, the lower the requested time, the more favorable your wait time will be. In other words, if the job is estimated to execute for one hour, a request for 24 hours will waste valuable turnaround time. The same job would take considerable less queue wait time with a requested limit of two hours.

There is a very useful tool available to all users that, together with the script manipulation, can cut down the queue wait time dramatically. The "showbf" command has many options; however, we will look at the standard command without any flags for this article.

```
habu% showbf

backfill window (user: 'jared' group: 'NA0101'
partition: ALL) Wed Aug 28 09:47:04

7 nodes available for       8:05:51
```

Number of Nodes (1 Node = 4 Processors)

Requested Time (in Hours)	1	4	8	16	32
.5	2	2	2	5858	18786
1	3	2	2	2	24987
3	2	154	2340	5628	825
6	18	123	1675	837	880
9	2	122	1576	1056	2
12	2	2	1156	2	74
15	2	123	28185	95	24958
18	2	1	27987	133	24839
21	2	2	85878	39	2
24	2	1	85970	123	145

Approximate Queue Wait Time (in Seconds)

Table 1. A two-week run of jobs from 1 to 32 nodes at all possible wall clock limits.

Looking at the output we see that there are currently seven nodes available for a time limit of a little over eight hours. Any job submitted that fits these parameters will gain an immediate reservation and will be executed immediately. What this command shows is actually the available space for backfill on the system, or which nodes are available for use on the system. This command is very useful in obtaining the fastest possible turnaround for your job.

Table 1 presents a 2-week run of jobs from 1 to 32 nodes at all possible wall clock limits. It is easy to see that while some jobs took hours to wait, under certain instances a much larger job took considerable less time. This is true because the job at the time of submission was able to take advantage of a spot in the backfill of the system.

Thus, with three simple steps one can obtain the fastest possible turnaround for a particular job.

1. First run "showbf":

```
habu% showbf
backfill window (user: 'jared' group:
'NA0101' partition: ALL) Wed Aug 28
09:47:04
7 nodes available for    8:05:51
```

   If the job is capable of finishing in the amount of time specified by the "showbf" command then adjust the run script accordingly. This will provide an immediate execution of the job.

2. If the "showbf"command returns no available time or the time available is not large enough for execution, then adjust the run script based on previous experience or estimation of approximate run-time. For example: If in the past your job took only three hours to execute with two nodes requested, then adjust the wall clock limit to four hours rather than from the queue maximum time limit. Alternatively, lower the number of requested nodes and raise the wall clock time.
3. Use Table 2 to predict your approximate queue wait time and adjust your script accordingly. Table 2 represents figures based on historical queue wait data over a five-month period.

While it may take a few minutes initially to make these changes, it will ultimately save a lot of valuable time and system load by making a few simple changes.

Number of Nodes (1 Node = 4 Processors)

Requested Time (in Hours)	1	4	8	16	32
.5	19	380	425	1460	5508
1	86	635	451	4085	6343
3	165	1004	2117	6368	15977
6	642	1712	13056	10290	34693
9	839	1823	15960	24130	76680
12	985	2354	12214	42235	56680
15	1132	2163	19120	51401	24958
18	1109	2032	27987	47470	98666
21	1257	2469	52015	86329	69057
24	1280	2829	87457	84962	129296

Approximate Queue Wait Time (in Seconds)

Table 2. Figures based on historical queue wait data over a five-month period.

HABU to MARCELLUS...continued from page 19

To realistically use more than 1 GB per task, you must run less than eight tasks per node or use the high-memory nodes. These nodes have 64 GB available per node.

Transferring files between MARCELLUS and the archive servers, JULES and VINCENT, works as on HABU. The rcp command or the Practical Supercomputing Toolkit archive command may be used between MARCELLUS and the archive servers. See the NAVO MSRC web pages for details.

See the sample batch script below for a side-by-side look at the changes that should be made for moving an application from HABU to MARCELLUS. The changes are highlighted.

HABU	MARCELLUS
#!/bin/ksh	#!/bin/ksh
# HABU	# Marcellus
#@ output = mytest.out	#@ output = mytest.out
#@ error = mytest.err	#@ error = mytest.err
#@ account_no = Project_Name	#@ account_no = Project_Name
#@ wall_clock_limit = 4:00:00	#@ wall_clock_limit = 4:00:00
#@ class = batch	#@ class = batch
#@ job_type = parallel	#@ job_type = parallel
#@ node_usage = not_shared	#@ node_usage = not_shared
#@ node = 16	#@ node = 8
#@ tasks_per_node = 4	#@ tasks_per_node = 8
#@ network.mpi = css0,shared,us	#@ network.mpi = csss,shared,us
#@ environment = \	#@ environment = \
MP_EUILIB=us; \	MP_EUILIB=us; \
MP_EUIDEVICE=css0;	MP_EUIDEVICE=csss;
#@ queue	#@ queue
#----------	#----------
# compile executable	# compile executable with optimization
mpxlf -O3 -qarch=pwr3 -qtune=pwr3 \ -bmaxdata:0x70000000 -o myprog \ myprog.f	mpxlf_r -O3 -qarch=pwr4 -qtune=pwr4 \ -bmaxdata:0x70000000 -o myprog \-q64 -o myprog myprog.f
# copy executable and any required	# copy executable and any required
# input files to your batch work	# input files to your batch work
# directory, located under the	# directory, located under the
# /scr GPFS filesystem.	# /scr GPFS filesystem.
cp $HOME/myprog /scr/$USER/myprog	cp $HOME/myprog /scr/$USER/myprog
cp $HOME/myinput/* /scr/$USER/	cp $HOME/myinput/* /scr/$USER/
# load and run parallel code on all	# load and run parallel code on all
# requested nodes under the poe	# requested nodes under the poe
# jobstarter command.	# jobstarter command.
cd /scr/$USER	cd /scr/$USER
poe ./myprog	poe ./myprog
# Archive any output from the job	# Archive any output from the job
# to your home directory. cp ./myoutput.file $HOME/myoutput.file	# to your home directory. cp ./myoutput.file $HOME/myoutput.file

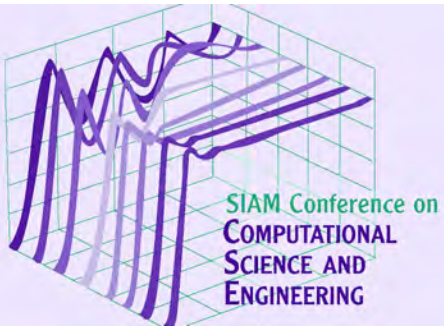
Upcoming Events

International Conference on High Performance Computing

December 18-21, 2002
Bangalore, India
www.hipc.org

SIAM Conference on Computational Science and Engineering

February 9-13, 2003
San Diego, CA
http://www.siam.org/meetings/cse03



2003 ACM Symposium on Applied Computing

March 9-12, 2003
Melbourne, FL
http://www.acm.org

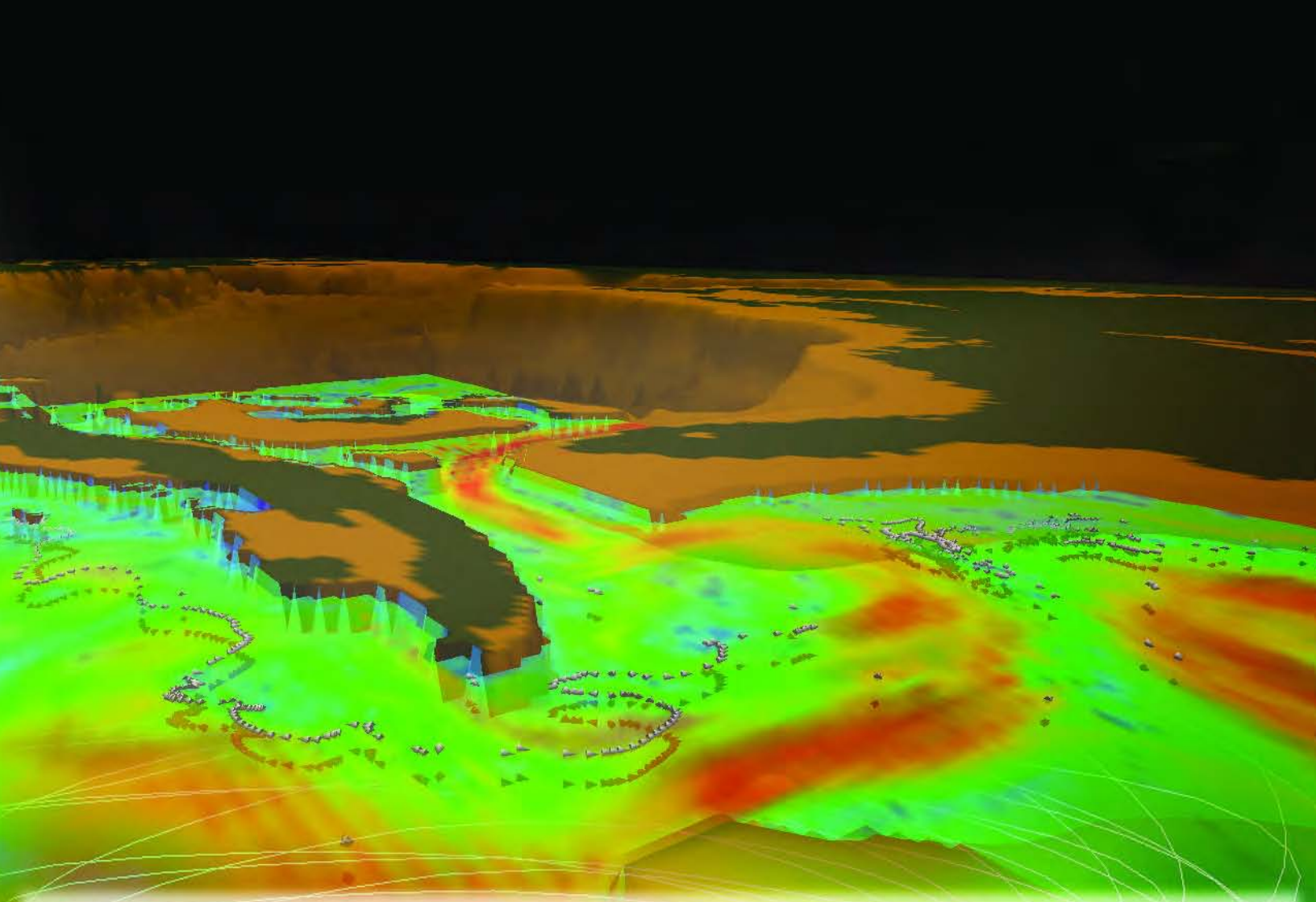
SIAM Conference on Mathematical and Computational Issues in the Geosciences (SIAG/GS)

March 17-20, 2003
Austin, TX
http://www.siam.org/meetings/gs03

Annual High Performance Computing and Communications Conference

25-27 March 2003
Newport, RI
www.hpcc-usa.org/genconf.html

ANNUAL HIGH PERFORMANCE COMPUTING AND COMMUNICATIONS CONFERENCE



Naval Oceanographic Office * MAJOR SHARED RESOURCE CENTER
1002 Balch Boulevard . Stennis Space Center, Mississippi . 39522